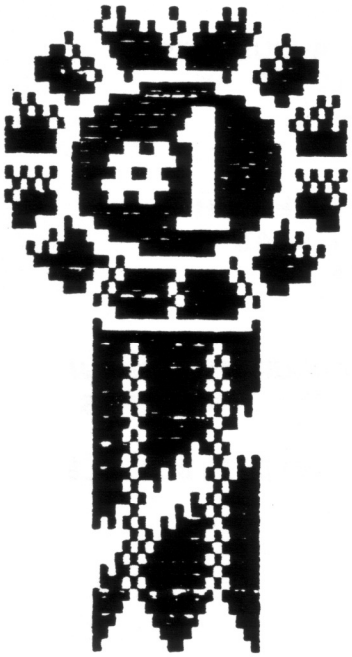
VZ-LINK MAGAZINE

NEW ZEALAND

VZ 300/200

COMPUTER USERS CLUB



ISSUE 39 AUGUST 89

COMMENT

Welcome to the second half of 1989!!!

I hope you are all able to make it to the next Workshop.Details below.
I have now received the SPO 256 Speech Processor Chip and hope to have my Synthesiser up and running in time for the meeting.

This issue Includes a very fine program written by Ken Clarke called "JUKEBOX",I am sure you will enjoy it.
We also have another item from Joe Leon,This time Joe has shown us how to add a disk utility to your Programs
I have included a index of issues 21 to 26 on the back page.
Next issue I will bring the Index fully upto date.

If anyone has the Computor bet program from issue 29 typed In,could you please give me a copy.

Please note that we are running out of items for the newsletter fast.So if you can help please do so.

If there is anything you have any queries for,please do not hesitate to give me a call;I am always happy to help you solve a problem.

MEETING 12 AUGUST

****
AUCKLAND VZ 300/200 COMPUTER USERS CLUB
****

VZ USERS-WORKSHOP/MEETING

WE WILL BE HOLDING A VZ USERS WORKSHOP ON:-

SATURDAY 12TH AUGUST 89
STARTING AT 10-30 AM

AT THE :-

OWAIRAKA BAPTIST CHURCH
119 RICHARDSON ROAD
OWAIRAKA

BRING YOUR FAVOURITE SOFTWARE ETC
BRING YOUR PROBLEM'S

A LIGHT LUNCH WILL BE PROVIDED,BUT PLEASE BRING A PLATE TO HELP

JUKE BOX

```
10 REM   *** THE JUKE BOX ***
20 REM  *** < KEN CLARKE > ***
60 CLS:PRINT"     <MAKE YOUR SELECTION>"
65 PRINT:PRINT"1>  TIE ME KANGAROO DOWN"
66 PRINT"2>  ADVANCE AUSTRALIA FAIR"
67 PRINT"3>  PUB WITH NO BEER"
68 PRINT"4>  BOTANY BAY"
69 PRINT"5>  SAILOR'S HORNPIPE"
70 PRINT"6>  BACH - SUITE IN B MINOR"
71 PRINT"7>  RUN RABBIT"
72 PRINT"8>  MEDLEY"
73 PRINT"9>  THE GREEN LEAVES OF SUMMER"
74 PRINT"10> SHE WORE A YELLOW RIBBON"
75 PRINT"11> CIELITO LINDO"
76 PRINT"12> CLICK GO THE SHEARS"
77 PRINT"13> THE JUKE BOX              ";
80 INPUTX
87 IFX=1,105
88 IFX=2,205
89 IFX=3,305
90 IFX=4,405
91 IFX=5,505
92 IFX=6,605
93 IFX=7,705
94 IFX=8,805
95 IFX=9,905
96 IFX=10,1005
97 IFX=11,1105
98 IFX=12,1205
99 IFX=13,105
100 GOTO80
105 CLS:PRINT:PRINT"            PLEASE WAIT..."
106 FORM=1TO300:NEXTM
107 CLS:PRINT@230,"TIE ME KANGAROO DOWN"
110 SOUND4,3;9,1;13,3;11,1;9,3;6,5;11,5
120 SOUND4,3;8,1;11,3;14,1;18,3;16,7;0,1
130 SOUND13,3;16,1;13,3;11,1;9,3;6,5;11,5
140 SOUND4,3;8,1;11,3;9,1;8,3;9,5
150 SOUND16,2;16,1;16,2;16,1;16,3;0,1
160 SOUND4,3;9,1;13,3;11,1;9,3;6,5;11,5
170 SOUND4,3;8,1;11,3;14,1;18,3;16,7;0,1
180 SOUND13,3;16,1;13,3;11,1;9,3;6,5;11,5
190 SOUND4,3;8,1;11,3;9,1;8,3;9,8
200 IFX=1,60
205 CLS:PRINT:PRINT"            PLEASE WAIT..."
206 FORB=1TO1000:NEXTB
207 CLS:PRINT@229,"ADVANCE AUSTRALIA FAIR"
210 SOUND11,4;16,4;11,4;8,4;11,4;16,5;16,2;16,4
220 SOUND20,4;18,4;16,4;15,4;16,4;18,8
230 SOUND11,4;16,4;11,4;8,4;4,4;11,5;11,2;11,4
240 SOUND20,4;18,4;16,4;15,4;13,4;11,8
250 SOUND11,4;13,5;15,2;16,4;13,4;11,5;8,2;8,4
260 SOUND11,4;13,4;16,4;21,4;20,4;18,8
270 SOUND11,4;13,5;15,2;16,4;13,4;11,5;16,2;16,4;18,4
280 SOUND20,5;16,2;18,5;15,2;16,8
290 SOUND20,4;21,4;20,4;18,4;16,4;15,4;13,4;11,4
295 SOUND16,4;20,5;16,2;18,5;15,2;16,8
300 IFX=2,60
```

by

Ken Clarke

```
305 CLS:PRINT:PRINT"            PLEASE WAIT..."
306 FORB=1TO1000:NEXTB
307 CLS:PRINT@231,"PUB WITH NO BEER"
310 SOUND18,2;18,4;17,4;20,4;18,4;13,5;10,2;8,4;7,4;8,4
320 SOUND15,6;15,2;15,2;13,4;17,4;15,4;13,6
330 SOUND8,2;8,2;8,4;10,4;11,4;10,6
340 SOUND18,2;18,2;18,4;17,4;20,4;18,4;13,5;10,2;8,4;7,4;8,4
350 SOUND15,6;15,2;15,2;13,4;17,4;15,4;13,6
360 SOUND8,2;10,2;11,4;10,4;8,4;6,6
370 SOUND18,2;18,2;18,4;17,4;20,4;18,4;13,5;10,2;8,4;7,4;8,4
380 SOUND15,6;15,2;15,2;13,4;17,4;15,4;13,6
390 SOUND8,2;10,2;11,4;10,4;8,4;6,6
400 IFX=3,60
405 CLS:PRINT:PRINT"            PLEASE WAIT..."
406 FORB=1TO1000:NEXTB
407 CLS:PRINT@235,"BOTANY BAY"
410 SOUND7,1;9,1;11,2;14,2;14,2;9,2;12,2;12,2;11,1;9,1;7,5;0,1
420 SOUND14,2;11,2;14,2;19,2;12,2;16,2;19,2;14,6;0,1
430 SOUND14,2;19,2;18,2;19,2;21,2;19,2;16,2;14,2;11,1;7,5;0,1
440 SOUND7,1;9,1;11,2;14,2;14,2;9,2;12,2;11,2;7,6;0,1
450 SOUND7,1;9,1;11,2;14,2;14,2;9,2;12,2;12,2;11,1;9,1;7,5;0,1
460 SOUND14,2;11,2;14,2;19,2;12,2;16,2;19,2;14,6;0,1
470 SOUND14,2;19,2;18,2;19,2;21,2;19,2;16,2;14,2;11,1;7,5;0,1
480 SOUND7,1;9,1;11,2;14,2;14,2;9,2;12,2;11,2;7,6
500 IFX=4,60
505 CLS:PRINT:PRINT"            PLEASE WAIT..."
506 FORB=1TO1000:NEXTB
507 CLS:PRINT@232,"SAILOR'S HORNPIPE"
510 SOUND23,1;20,1;16,1;16,1;11,1;16,1;16,1;20,1;16,1;20,1;23,1
520 SOUND21,1;20,1;21,1;18,1;18,1;13,1;18,1;18,1;21,1;18,1
530 SOUND21,1;25,1;23,1;21,1;20,1;16,1;16,1;11,1;16,1;16,1;20,1
540 SOUND16,1;20,1;23,1;21,1;20,1;21,1;20,1;21,1;25,1;23,1
550 SOUND21,1;20,1;16,1;16,1;16,1;0,1
560 SOUND23,1;20,1;16,1;16,1;11,1;16,1;16,1;20,1;16,1;20,1;23,1
570 SOUND21,1;20,1;21,1;18,1;18,1;13,1;18,1;18,1;21,1;18,1
580 SOUND21,1;25,1;23,1;21,1;20,1;16,1;16,1;11,1;16,1;16,1;20,1
590 SOUND16,1;20,1;23,1;21,1;20,1;21,1;20,1;21,1;25,1;23,1
595 SOUND21,1;20,1;16,1;16,1;16,1;0,1
600 IFX=5,60
605 CLS:PRINT:PRINT"             PLEASE WAIT..."
606 FORB=1TO1000:NEXTB
607 CLS:PRINT@228,"BACH - SUITE IN B MINOR"
610 SOUND27,2;30,1;27,1;22,2;27,1;22,1;18,2;22,1;18,1;15,4;10,1
620 SOUND15,1;18,1;15,1;17,1;15,1;17,1;15,1;14,1;17,1;20,1;17,1
630 SOUND18,2;15,2;27,2;30,1;27,1;22,2;27,1;22,1;18,2;22,1;18,1
640 SOUND15,4;18,2;18,2;18,2;18,2;27,2;18,2;18,2;17,2;22,2;22,2
650 SOUND22,2;22,2;30,2;22,2;22,2;21,2;17,1;22,1;22,1;25,1;22,1
660 SOUND24,1;22,1;24,1;22,1;21,1;24,1;27,1;24,1;25,1;24,1;25,1
670 SOUND24,1;22,1;25,1;22,1;21,1;22,1;27,1;22,1;21,1;22,1;29,1
680 SOUND22,1;21,1;22,1;30,1;22,1;21,1;22,1;30,1;29,1;27,1;29,1
690 SOUND25,1;24,1;22,1;25,2;24,2;22,4;0,4
700 IFX=6,60
```

```
705 CLS:PRINT:PRINT"                PLEASE WAIT..."
706 FORB=1TO1000:NEXTB
707 CLS:PRINT@235,"RUN RABBIT"
710 SOUND16,4;25,2;23,2;16,4;25,2;23,2;15,4;15,4;15,6
720 SOUND15,4;25,2;23,2;15,4;25,2;23,2;16,4;16,4;16,6;28,4;28,4
730 SOUND28,4;28,4;27,2;25,2;24,2;25,2;18,6;18,2;16,4;25,2;23,2
740 SOUND16,4;25,2;23,2;28,4;27,4;25,4;23,4
750 SOUND16,4;25,2;23,2;16,4;25,2;23,2;15,4;15,4;15,6
760 SOUND15,4;25,2;23,2;15,4;25,2;23,2;16,4;16,4;16,6;28,4;28,4
770 SOUND28,4;28,4;27,2;25,2;24,2;25,2;18,6;18,2;16,4;25,2;23,2
780 SOUND16,4;25,2;23,2;28,4;28,4;28,6
800 IFX=7,60
805 CLS:PRINT:PRINT"                PLEASE WAIT..."
806 FORB=1TO1000:NEXTB
807 CLS:PRINT@237,"MEDLEY"
810 SOUND16,1;20,1;23,3;23,3;23,3;23,3;23,6;28,1;23,1;0,1
815 SOUND20,1;21,1;23,4;23,4;21,4;18,4;16,8;0,1
820 SOUND20,5;25,5;25,4;25,1;25,5;29,4;27,1;25,5;24,4;22,1;20,6
830 SOUND0,2;16,2;20,2;21,2;23,6;0,1;16,2;20,2;21,2;23,6;0,1
835 SOUND16,2;20,2;21,2;23,4;20,4;16,4;20,4;18,6;0,1
840 SOUND21,2;20,2;18,2;16,4;16,4;20,4;23,2;23,2;21,4;21,2;20,2
845 SOUND21,2;23,4;20,4;16,4;18,4;16,8;0,1
850 SOUND20,2;24,2;24,2;24,2;22,1;20,1;27,6;27,1;26,1;24,2
855 SOUND24,2;24,2;22,1;20,1;27,6;27,1;26,1;24,2;26,1;27,1;26,2
860 SOUND24,2;22,2;19,2;15,2;0,1
870 SOUND0,1;23,5;23,6;28,3;28,3;27,3;27,7;21,5;21,6;25,3;25,3
875 SOUND23,3;23,7;20,5;25,6;23,3;23,3;21,3;21,7;21,5;20,5
880 SOUND18,5;25,5;23,3;23,7
900 IFX=8,60
905 CLS:PRINT:PRINT"                PLEASE WAIT..."
906 FORB=1TO1000:NEXTB
907 CLS:PRINT@227,"THE GREEN LEAVES OF SUMMER"
910 SOUND16,2;21,6;23,1;21,1;20,2;16,4;16,2;21,6;23,1;24,1;23,2
920 SOUND19,4;19,2;24,6;23,1;24,1;26,2;21,4;26,2;27,6
930 SOUND23,1;27,1;28,6;28,1;29,1;31,6;29,1;28,1;26,2;21,4
940 SOUND26,1;28,1;29,6;28,1;26,1;24,2;19,4;24,1;26,1;28,6
950 SOUND26,1;24,1;23,2;26,4;23,2;21,4;24,2;21,2;23,6
960 SOUND21,2;21,6;23,1;21,1;20,2;16,4;16,2;21,6
970 SOUND23,1;24,1;23,2;19,4;19,2;24,6;23,1;24,1;26,2;21,4
980 SOUND26,2;27,6;23,1;27,1;28,6;28,1;29,1;31,6;29,1;28,1;26,2
990 SOUND21,4;26,1;28,1;29,6;28,1;26,1;24,6;24,1;26,1;28,6
995 SOUND26,1;24,1;23,2;26,4;23,2;21,6;23,1;20,1;21,8
1000 IFX=9,60
1005 CLS:PRINT:PRINT"                PLEASE WAIT..."
1006 FORB=1TO1000:NEXTB
1007 CLS:PRINT@228,"SHE WORE A YELLOW RIBBON"
1010 SOUND23,4;20,1;23,4;20,1;23,1;25,1;23,1;21,1;20,1;23,4
1015 SOUND23,1;16,1;16,1;16,1;18,1;20,1;20,1;20,1;20,1;20,1
1020 SOUND18,1;18,1;18,1;20,1;18,4;0,1;23,4;20,1;23,4;20,1
1025 SOUND23,1;25,1;23,1;21,1;20,1;23,3;23,1;16,1;16,1;16,1
1030 SOUND18,1;20,1;20,1;20,1;16,1;18,3;16,1;15,1;16,1;0,1
1035 SOUND28,1;27,1;25,6;28,1;25,1;23,6;16,1;15,1;16,1;16,1
1040 SOUND16,1;18,1;20,3;20,1;16,1;18,3;16,1;15,1;16,1;0,1
1045 SOUND28,1;27,1;25,6;28,1;25,1;23,6;20,1;18,1;16,1;16,1
1050 SOUND16,1;18,1;20,3;20,1;16,1;18,3;16,1;15,1;16,5;0,4
1100 IFX=10,60
```

```
1105 CLS:PRINT:PRINT"                    PLEASE WAIT..."
1106 FORB=1TO1000:NEXTB
1107 CLS:PRINT@234,"CIELITO LINDO"
1110 SOUND26,1;26,1;23,2;25,1;21,1;26,1;26,1;23,2;25,1;21,1
1115 SOUND26,1;26,1;23,2;25,1;21,1;19,1;16,5;25,1;25,1;25,2
1120 SOUND23,1;21,1;19,1;19,1;16,2;18,1;19,1;21,1;21,1;21,2
1125 SOUND19,1;18,1;16,1;14,5;30,3;28,2;26,1;23,6;28,3;28,2
1130 SOUND26,1;30,1;26,4;21,1;23,2;21,1;23,1;23,1;21,1;31,1;31,1
1135 SOUND28,2;25,1;21,1;23,1;23,1;21,2;19,1;21,1;18,1;16,1;14,8
1200 IFX=11,60
1205 CLS:PRINT:PRINT"                    PLEASE WAIT..."
1206 FORB=1TO1000:NEXTB
1207 CLS:PRINT@231,"CLICK GO THE SHEARS"
1210 SOUND19,4;19,2;17,2;15,2;15,2;19,2;22,2;27,4;27,2;26,2;24,7
1215 SOUND22,4;22,2;24,2;22,4;20,2;19,2;17,4;17,2;19,2;17,7
1220 SOUND19,4;19,2;17,2;15,4;19,2;22,2;27,4;27,2;26,2;24,7;29,2
1225 SOUND27,2;26,2;24,2;22,2;20,2;19,2;17,2;15,2;27,2;27,2;27,2
1230 SOUND27,6;0,1;29,4;29,2;26,2;22,4;29,4;27,4;31,4;27,6
1235 SOUND24,4;24,2;26,2;27,4;27,2;24,2;22,4;27,4;17,5;17,2
1240 SOUND19,2;19,2;19,2;17,2;15,4;19,2;22,2;27,2;27,2;27,2;26,2
1245 SOUND24,5;24,2;29,2;27,2;26,2;24,2;22,2;20,2;19,2;17,2;15,4
1250 SOUND27,2;27,2;27,6
1300 IFX=12,60ELSE105
```

A FREE DISC copy can be arranged at a Workshop Meeting, or through Peter.

vz dos

Last edition we looked at the physical side of the VZ disk drive system, and then at the software side of it. We saw how each sector was structured, and how a sector was the smallest 'updatable' unit on the disk (as far as the existing software goes). This month we look at the directory and the track map of a disk.

The VZ DOS (Disk Operating System) uses track number 0 to store what is known as the directory and the track map Quite obviously, if the disk is to be used to store individual files, as all PC systems are, then some record must be kept of where each file is physically located on the disk The name and type of each file is stored with this information, as well as some information about where the file normally resides in memory (its starting and ending address) Lastly, the system needs to keep a record of what areas of the disk are currently being used to store files, in order that when a new file is to be saved, the system can use sectors not already holding important data. This information is stored in the track map.

The VZ DOS uses track 0, sectors 0 to 14 to store the directory, and sector 15 to store the track map. Each of the 15 sectors used for the directory can hold up to 8 entries (each entry uses 16 bytes - 8x16=128 bytes per sector). Each entry in the directory contains a file type flag filename, start track, start sector, start address, and end address

These are set out as follows:

File Type	1 byte
Delimitor	1 byte
Filename	8 bytes
Start track	1 byte
Start sector	1 byte
Start address	2 bytes (Hi-Lo form)
End address	2 bytes (Hi-Lo form)

The file type flag is simply the ASCII value of the characters which appear when you do a DIR. That is, a BASIC program, which is a 'T' file, has a type flag of 54H (84 decimal - the ASCII code for a T), a binary program 42H ('B') etc. This flag can be any value you like (you could have X files if you wanted!) but you wouldn't be able to use the ordinary DOS commands (we'll explore these later).

The Delimitor is a fancy name for the byte separating the file type and filename. This is usually 3AH (a colon :), but new DOSs being produced (such as Russell Harrison's) use different values to indicate that the file is to be executed when the system is reset. Once again, this code can be anything you want it to be!

The next eight bytes store the filename in ASCII format. This is what limits filenames to eight bytes!

The next two bytes tell the system the first sector in the 'chain of sectors used to store the file, i.e. where the program starts on the disk. First the track number is given, followed by the sector number. We'll see how the system finds the rest of the program in a moment.

Next comes the starting address of the file in memory. When the file is loaded (not data files) it loads into memory starting at this location. The address is in high-low byte form since it is a 16-bit address (if you don't understand this, see Vol.1 No.1). The ending address of the file comes next, and is in the same format.

Let's follow the system through as it would normally load a BASIC program for example. The directory is first read, starting at track 0 sector 0, until the filename specified by the user is found in one of the entries. If it is not found, the ?FILE NOT FOUND error message is generated. If it is found, the system copies the start track/sector and start/end address into RAM, after checking that the file type matches the type wanted. It then moves to the track/sector specified by the start track and start sector bytes in the directory entry, and reads in this sector (128 bytes) starting at the starting address. Only the first 126 bytes are actually part of the program; the last two bytes of each sector tell the system where the next sector in the chain is to be found. This process of reading sectors continues until the last two bytes of a sector are zero, indicating that this is the last sector to be read. By using this system, the sequence of sectors holding a particular file may be spread all over the disk, but the system always knows where the next one will be!

Now let's turn our attention to the track map. This is stored in track 0 sector 15. Each sector on the disk has a corresponding bit in the track map. If the bit representing a sector is 0, then that sector is currently unused. If the bit is a 1 then the sector is being used. Since there are 16 sectors per track, and eight bits per byte, two bytes of the track map are needed to map one full track (16 sectors).

Since there are 39 tracks other than the directory track used for storing files, the track map is 78 bytes long. The remaining 50 bytes of track 0 sector 15 are unused by the DOS (good for storing serial numbers?). The format of the track map is as shown below:

VZ DOS TRACK/SECTOR MAP

Byte 1:	Bit 0 (LSB)	-	Track 1	Sector 0
" :	Bit 1	-	"	Sector 1
" :	Bit 2	-	"	Sector 2
" :	Bit 3	-	"	Sector 3
" :	Bit 4	-	"	Sector 4
" :	Bit 5	-	"	Sector 5
" :	Bit 6	-	"	Sector 6
" :	Bit 7	-	"	Sector 7
Byte 2:	Bit 0	-	"	Sector 8
" :	Bit 1	-	"	Sector 9

(this pattern repeats up to Track#39 Sector#15)

We can now appreciate what goes on when we load, save, erase, rename, DIR and STATUS a disk. When saving, the system looks for empty sectors in the track map (which may be scattered over the disk) and uses these to save our file in. It creates a new entry in the directory for the file, and updates the track map. When a file is erased, the sectors that were being used by the file are all 'marked' as being unused (with 0 bits) in the track map, and the entry for the file in the directory has a '1' placed in the file type flag; the rest of the entry remains intact (this allows new DOS versions to 'unerase' a file, providing the file type is supplied).

Renaming a file simply rewrites the filename in the particular directory entry. A DIR simply displays the first 10 bytes of each directory entry on the screen, while a STATUS counts the number of unused sectors in the track map and converts this to memory size (in K) and sectors (records free)

Now that we understand how file 'book keeping' is done by the VZ DOS, we can go on to write our own DOS routines. Next month we present a small utility program for editing sectors, and look at the DOS Vector table.

P.O.BOX 1972 C.P.O AUCKLAND N.Z

(C) AUCKLAND VZ 300/200 COMPUTER USER'S CLUB 1986

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT..
CONTRIBUTED PROGRAMS & ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT THEY ARE FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.
COPYRIGHT IS ALWAY'S RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR THE BREECHING OF COPYRIGHT CAUSED BY THE PUBLICATION OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME IF YOU HAVE PROBLEMS
PHONE 761-451

DISK UTILITY BY JOE LEON . . .

```
V-ZED DISK UTILITY
E - ERA  FILE   R - REN  FILE
L - LOAD DATA   S - SAVE DATA
D - DRIVE [ ]   P - SAVE PROG
I - DIRECTORY   M - MEM  PEEK
N - INIT DISK   Q - MAIN MENU
DISPLAY DIRECTORY
SPACE TO START  Q-QUIT
```

```
SPACE START/STOP   Q-QUIT
31477  69  E
31478  82  R
31479  65  A
31480  34  "
31481  49  1
31482  50  2
31483  51  3
31484  52  4
31485  53  5
31486  54  6
31487  55  7
31488  56  8
31489  34  "
31490  58  :
PRESS RETURN FOR MENU
```

```
10 GOTO46
12 ERA"12345678":RETURN
14 REN"12345678","12345678":RETURN
16 SAVE"12345678":RETURN
38 '"DO NOT ALTER LINES 10-16
40 FORPO=0TO7:POKEAD+PO,ASC(MID$(N$+"        ",PO+1,1)):NEXT
42 RETURN
44 :
46 CLS:POKE30744,96:COLOR,0:POKE30862,80:POKE30863,52
48 DR=1:DR$="D":S$=" ------------------------------"
50 DF$="SELECT DESIRED FUNCTION"
52 SP$="                              ":REM 30 SPACES
56 GOTO300
58 :
60 PRINT@482,"SPACE TO START  Q-QUIT";:SOUND30,1
62 A$=INKEY$:A$=INKEY$:IF A$=""THEN62ELSEX=USR(X)
64 IFA$=" "THENGOSUB75:RETURNELSEIFA$="Q"THEN300ELSE62
68 :
70 PRINT@418,SP$:RETURN
75 PRINT@480,SP$:RETURN
80 GOSUB70:PRINT@418,"DISPLAY DIRECTORY  ";:GOSUB60
82 CLS:DIRA:STATUSA
84 :
86 PRINT@482,"PRINT DIRECTORY - Y/N";:SOUND30,1
88 R$=INKEY$:R$=INKEY$:IFINKEY$=""THEN88ELSEX=USR(X)
90 IFR$="N"THEN300ELSEIFR$="Y"THEN92ELSE88
92 IF INP(0)=254 THENCLS:POKE30876,1:DIR:STATUS:GOTO300
94 :
96 PRINT@482,"PRINTER NOT OPERATIONAL";:SOUND30,1
98 SOUND0,9;0,5:GOTO300
98 :
100 GOSUB70:PRINT@418,"ERA FILENAME  ";:SOUND30,1
102 INPUTN$:AD=31481:GOSUB40:GOSUB60:GOSUB12:GOTO82
108 :
110 GOSUB70:PRINT@418,"OLD FILENAME  ";:SOUND30,1
112 INPUTN$:AD=31501:GOSUB40
114 PRINT@450,"NEW FILENAME  ";:SOUND30,1
116 INPUTN$:AD=31512:GOSUB40:GOSUB60:GOSUB14:GOTO 82
118 :
120 GOSUB70:PRINT@444,"N"
122 PRINT@418,"UPDATE PROGRAM Y/N  ";:SOUND30,1:INPUTN$
124 IFN$="N"THEN UP=0 ELSEIFN$="Y"THEN UP=1
128 :
130 GOSUB70:PRINT@418,"SAVE PROGRAM  ";:SOUND30,1:INPUTN$
132 SOUND30,1:IFUP=0THENGOSUB60:GOTO140
134 GOSUB 60
136 GOSUB70:PRINT@418,"ERASING PROGRAM  ";N$:SOUND30,1
138 AD=31481:GOSUB40:GOSUB12
140 GOSUB70:PRINT@418,"SAVING PROGRAM  ";N$:SOUND30,1
142 AD=31533:GOSUB40:GOSUB16:GOTO 82
```

DISK UTILITY CONT. . . . . .

```
190 GOSUB70:PRINT@418,"▌ INITIALIZE DISK ▌";:GOSUB 60
192 PRINT@482,"▌ PLEASE WAIT 15 SECONDS ▌";:SOUND30,1
194 INIT:GOTO 80
198 :
200 CLS:POKE30777,1:PRINT"▌ SPACE START/STOP Q-QUIT ▌";
210 FORL= 31477 TO 31847:PRINTTAB(9)L;PEEK(L);CHR$(PEEK(L))
220 IFINKEY$=" "THEN230ELSEIFINKEY$="Q"THEN240
230 IFINKEY$=""THEN220ELSENEXT
240 PRINT@483,"▌ PRESS RETURN FOR MENU ▌";
245 POKE30777,1:SOUND30,1
250 IFINKEY$=CHR$(13)THEN300ELSE250
295 :
300 CLS:POKE30777,1:COLOR3:PRINTS$
305 PRINT"  ▌ V-ZED DISK UTILITY ▌ ":PRINTS$
310 PRINT"  E - ERA  FILE  R - REN  FILE":PRINT
315 PRINT"  L - LOAD DATA  S - SAVE DATA":PRINT
320 PRINT"  D - DRIVE [ ]  P - SAVE PROG":PRINT
325 PRINT"  I - DIRECTORY  M - MEM  PEEK":PRINT
330 PRINT"  N - INIT DISK  M - MAIN MENU":PRINTS$
335 GOSUB390:PRINT@418,DF$:SOUND30,1:POKE30777,35
345 :
350 D$=INKEY$:D$=INKEY$:IF D$=""THEN350ELSEX=USR(X)
355 IFD$="E"THEN100ELSEIFD$="R"THEN110
360 IFD$="L"THEN350ELSEIFD$="S"THEN350ELSEIFD$="$"THEN80
365 IFD$="I"THEN190ELSEIFD$="M"THEN350
370 IFD$="&"THEN120ELSEIFD$="?"THEN200
375 IFD$="D"THENDR=1+ABS(DR=1):DR$=CHR$(240+DR):GOSUB390
380 GOTO350
385 :
390 PRINT@237,DR$:SOUND25,1:DRIVE DR:RETURN
395 :
```

WARNING - SAVE PROGRAM BEFORE RUNNING IT !!!

THIS DOS UTILITY FEATURES THE BASIC DISK ROUTINES WHICH COULD BE INCORPORATED IN YOUR OWN PROGRAMS AND MOST OF THE ROUTINES PRESENTED HERE WERE FEATURED IN DISKLIST IN ISSUE #15, PAGES 10-13 OF THE HUNTER VALLEY VZ JOURNAL. MAIN REASON OF UTILITY IS TO PROVIDE DISK USERS ABILITY TO USE VARIABLE FILENAMES.

THE EASIEST WAY TO DO THAT IS BY POKING THE FILENAMES INTO RESERVED LINES AT START OF PROGRAM AND FOR THAT REASON LINES AT START OF PROGRAM MUST NOT BE CHANGED BY A SINGLE CHARACTER OR SPACE AS IT WILL CORRUPT THE PROGRAM.

PROVISION HAS BEEN MADE FOR YOU TO CHANGE THEM AS YOU MAY WANT TO ADD FUNCTIONS LIKE DISK SAVE/LOAD OF DATA FILES, ETC. THE INBUILT MEM PEEK ROUTINE PROVIDES THE MEANS FOR YOU TO ALTER LINES 10-16 OR ADD MORE LINES WITH EXTRA DISK FUNCTIONS.

TWO SCREEN DUMPS ARE SHOWN, ONE IS OF THE MENU PRIOR TO SELECTING DIRECTORY WHILE THE OTHER SHOWS MEMORY PEEK ROUTINE IN ACTION. IN ALL FUNCTIONS A QUIT OPTION HAS BEEN PROVIDED.

ALL THE ROUTINES CAN BE RELOCATED TO SUIT YOUR PROGRAMS PROVIDED YOU ADJUST THE THEN, GOTO, GOSUB AND THEIR ASSOCIATED LINE NUMBERS. THE TWO ROUTINES WHICH MAKE IT POSSIBLE TO USE VARIABLE FILENAMES ARE MEM PEEK AT LINES 200-250 AND FILNAME POKE ROUTINE IN LINES 40-42. AND NOW TO THE PROGRAM ITSELF.

DISK UTILITY CONT. . . . . .

YOU'LL NOTE THAT LINES 12, 14, 16 AND 40-42 ARE ROUTINES. ALSO THE NUMBERS 1 TO 8 ARE BETWEEN THE QUOTES AND SIMPLY REPRESENT THE 8 FILENAME SPACES AND IT'S WHERE THE VARIABLE FILENAMES ARE POKED INTO.

IN CASE THERE ARE LESS THAN 8 CHARACTERS IN FILENAME THEN THE ROUTINE AT LINE 40 PADS IT OUT WITH SPACES TO MAKE UP THE 8 CHARACTERS. SO REGARDLESS WHAT FILENAME IS BETWEEN THE QUOTES, THE NEW FILENAME OVERWRITES IT COMPLETELY.

USING DISK UTILITY :-

ERA FILE - THIS OPTION IS USED TO ERASE ANY DISK FILE OF YOUR CHOICE. WHEN REQUESTED ENTER FILENAME TO BE ERASED. PRESS SPACE TO START, Q TO QUIT. ROUTINE AT LINE 12 IS USED.

REN FILE - THIS OPTION IS USED TO RENAME ANY DISK FILE OF YOUR CHOICE. WHEN REQUESTED ENTER OLD FILENAME AND NEW FILENAME. PRESS SPACE TO START, Q TO QUIT. ROUTINE AT LINE 14 IS USED.

LOAD DATA - THIS OPTION HAS BEEN LEFT UP TO YOU.

SAVE DATA - THIS OPTION HAS BEEN LEFT UP TO YOU.

DRIVE 1-2 - THIS OPTION IS USED TO SELECT DRIVE 1 OR 2. PRESSING 'D' WILL TOGGLE BETWEEN DRIVE 1 AND 2 WITH MENU DISPLAY BEING UPDATED AS WELL.

SAVE PROG - THIS OPTION IS USED TO SAVE/UPDATE THIS PROGRAM. THE ROUTINE FOR SAVING IS IN LINE 16 AND IN UPDATING ERASE ROUTINE IS USED BEFORE SAVE.

DIRECTORY - THIS OPTION IS USED TO DISPLAY DIRECTORY AND STATUS. LINE 82 CONTAINS THE COMMANDS DIRA AND STATUSA AND ARE FUNCTIONS AVAILABLE IN D.MITCHELL'S EXT.DOS. ADJUST LINE 82 ACCORDING TO YOUR DOS DISK COMMANDS.

MEM PEEK - THIS OPTION IS USED TO ALTER LINES 10 TO 16 AND OR ADD EXTRA LINES . USE LINE NUMBERS BETWEEN 18 AND 36 SO THEY FOLLOW LINE 16. TO USE ROUTINE WE'LL USE ERASE ROUTINE AS EXAMPLE.

IF YOU STUDY THE RIGHT SCREEN DUMP ON PAGE 10 YOU'LL SEE THE MEMORY LOCATIONS OF LINE 12 STARTING WITH THE WORD ERA AND ENDING WITH A COLON. THE MEMORY LOCATION WE'RE AFTER IS THE ONE FOLLOWING OPENING QUOTE WHICH IS 31481.

IF WE LOOK NOW AT LINE 102 WE'LL SEE THAT AD=31481 WHICH IS THE START ADDRESS OF FIRST CHARACTER OF FILENAME. AFTER START ADDRESS IS DECLARED GOSUB40 POKES FILENAME INTO LINE 12. GOSUB 60 IS THE SPACE TO START 'Q' TO QUIT ROUTINE. GOSUB 12 DOES THE ACTUAL ERASING OF FILENAME AND THEN DISPLAYS DIRECTORY VIA ROUTINE IN LINE 82.

IF YOU ALTER OR ADD MORE LINES DO NOT FORGET TO UPDATE THE MEMORY ADDRESSES IN LINES CONTAINING AD=XXXXX. ALSO THE END ADDRESS IN LINE 210 WILL HAVE TO BE CHANGED AS IT ONLY DISPLAYS MEMORY CONTENTS UP TO LINE 16.

INIT DISK - THIS OPTION IS USED TO INITIALISE YOUR DISKS AND IS PROVIDED WITH A QUIT OPTION.

MAIN MENU - THIS OPTION IS USED TO RETURN TO OTHER MENU/S.

SUPER PREDICTIONS

```
10 CLS
14 PRINT"  ----------------------------"
15 PRINT" =======[illegible]=======
16 PRINT"  ----------------------------"
20 PRINT" THE PURPOSE OF THIS PROGRAM IS"
25 PRINT" TO ATTEMPT TO PREDICT THE"
30 PRINT" AMOUNT OF SUPERANUATION YOU"
35 PRINT" MAY COLLECT AFTER A GIVEN"
40 PRINT" LENGTH OF TIME"
45 PRINT:PRINT" NO RESPONSIBILITY IS ACCEPTED"
50 PRINT" FOR ANY INCORRECT PREDICTION"
55 PRINT:PRINT" WHEN RUNNING HOLD DOWN SPACE"
60 PRINT" BAR FOR PAUSE IN PROGRAM"
65 PRINT"  [illegible]"
70 A$=INKEY$:A$=INKEY$
75 IF A$="C"THEN 80 ELSE 70
80 CLS
90 PRINT"  ----------------------------"
95 PRINT" =======[illegible]=======
100 PRINT"  ----------------------------"
110 PRINT
115 PRINT" EXISTING ACCUMALATED FUNDS (IF"
116 INPUT" ANY) IN $                        ";G
120 INPUT" PERSONAL MONTHLY INPUT IN $     ";A
130 PRINT" COMPANY MONTHLY INPUT (AFTER    TAX) IN $"
131 INPUT B
135 PRINT" ANTICIPATED ANNUAL SALARY "
140 INPUT" INCREASE IN A PERCENTAGE (%)    ";C
145 PRINT" ANTICIPATED ANNUAL INTEREST"
150 INPUT" RATE IN A PERCENTAGE (%)        ";D
160 INPUT" NUMBER OF YEARS TO RUN          ";E
163 CLS:PRINT TAB(5);"YEAR 1"
165 FORY=1TO E
170 FORX=1TO12
180 G=(G+A+B)*(1+(D/1200))
190 PRINT TAB(10);G
200 NEXT
210 FORZ=1TO1000:NEXT
212 A$=INKEY$:A$=INKEY$
213 IF A$=" "THEN GOTO212
215 SOUND12,4:SOUND16,4:SOUND12,5
220 A=A*(1+(C/100))
230 B=B*(1+(C/100))
235 IF Y+1>E THEN260
240 PRINT TAB(5);"YEAR";Y+1
250 NEXT
260 PRINT:PRINT"  PRESS R TO RERUN, E TO EXIT"
270 A$=INKEY$
280 A$=INKEY$:IF A$="R" GOTO 80
290 IF A$="E" GOTO 500
300 GOTO 270
500 CLS:END
```

NEWSLETTER INDEX
ISSUES 21-26

ISSUE 21..

24 BIT I/O PROGRAMMABLE PERIPHERAL INTERFACE BOARD (BOB KITCH)
PROGRAM-MULTIPLY (KEITH WRIGHT)
ASSEMBLY LANGUAGE-PART 6
PROGRAM-DISK DIRECTORY PRINTER (KEITH WRIGHT)
VZ TECH BULLETIN-PRINTING LOWER CASE AND CONTROL CHARACTERS
VZ TECH BULLETIN-OBTAINING COLOUR ON THE VZ
REVIEW-'DATA' (P HILL)
PROGRAM-ROBO COP (DANNY TODD)
HOW TO WRITE ADVENTURE GAMES-PART 1 (SCOTT LE BRUN)
USING HEAD CLEANING DISK (JOE LEON)

ISSUE 22..

WRITING ADVENTURE GAMES-PART 2 (SCOTT LE BRUN)
ASSEMBLY LANGUAGE-PART 7
VZ TECH BULLETIN-PROGRAMMING VZ JOYSTICKS
PROGRAM-SLOT MACHINE (KEITH WRIGHT)
NEWSLETTER INDEX-ISSUE 1 TO 15

ISSUE 23

PROGRAM-CHEQUE BOOK (SANDIE MCSWEENEY)
ASSEMBLY LANGUAGE-PART 8
INSTALLING A 32K STATIC RAM IN VZ 200 (JOHN MASON)
POINTERS (?)
PROGRAM-BATTLESHIPS (GORDON BROWELL)
IMPROVING VZ SOUND (JOE LEON)
NEWSLETTER INDEX ISSUE 16-20

ISSUE 24

PROGRAM-NAVIGATION CALCULATOR (KEITH WRIGHT)
REVIEW-PLANET PATROL (BRENDA HILL)
JOYSTICK MODIFICATION (PETER PATTERSON)
PROGRAM-MATHS TESTER (BROOKE SPONG)
SHIFT LOCK UPDATE (JOE LEON)
MAJOR M/LIST UPDATE (JOE LEON)
GUIDELINES TO CAREFUL COMPUTING
SCHEMATIC-QUICKSHOT II JOYSTICK

ISSUE 25

ASSEMBLY LANGUAGE (SCOTT LE BRUN)
ENHANCING VZ BASIC (LARRY TAYLOR) PART 1
PROGRAM-MATHS (WHITTAKER HAMILTON)
REVIEW-VZ DATA LOGGER ETI-1612 (SCOTT LE BRUN)
DISKLIST UPDATE (JOE LEON)

ISSUE 26

PUT/GET*3 (ROBERT QUINN)
ASSEMBLY LANGUAGE An Overview of the VZ ROM (DANIEL AYERS)
ENHANCING VZ BASIC CONTINUED (LARRY TAYLOR)
BLOCK TRANSFERS (CHRIS HOBROUGH)